

The drawmatrix package

Elmar Peise
peise@aices.rwth-aachen.de

2016/08/25 v1.2.0

Abstract

drawmatrix provides macros to visually represent matrices. Various options allow to change the visualizations, e.g., drawing rectangular, triangular, or banded matrices.

Contents

1	Introduction	1
2	Drawing Matrices	1
2.1	Size	2
2.2	Shape	2
2.2.1	Triangular and Trapezoidal Matrices	2
2.2.2	Banded Matrices	3
2.2.3	Diagonal Matrices	3
2.3	Colors and Style	3
2.4	The Bounding Box	4
2.5	Coordinate system transformations	4
2.6	Position of the Label and Baseline	5
3	Changing Defaults	5
4	Externalization	5
5	Implementation	6
5.1	Package: TikZ	6
5.2	If for externalization	6
5.3	Key Declarations and Defaults	6
5.4	User Macros	8

1 Introduction

In many situations, visual representations of matrices facilitate the understanding of linear algebra properties, relations, and operations enormously. This package provides simple tools to bring such representations to L^AT_EX. For instance,

$$\begin{array}{|c|} \hline \diagdown \\ \hline A \\ \hline \end{array} \begin{array}{|c|} \hline X \\ \hline \end{array} + \begin{array}{|c|} \hline X \\ \hline \end{array} \begin{array}{|c|} \hline \diagdown \\ \hline B \\ \hline \end{array} = \begin{array}{|c|} \hline C \\ \hline \end{array}$$

is typeset as follows:

```

 $\$$ 
\drawmatrix[upper]A \;
\drawmatrix[width=.5]X +
\drawmatrix[width=.5]X \;
\drawmatrix[upper, size=.5, bbox height=1]B =
\drawmatrix[width=.5]C
 $\$$ 

```

2 Drawing Matrices

`\drawmatrix` `\drawmatrix[options]{label}` draws a matrix labeled *label*: `\drawmatrix A` produces $\begin{array}{|c|} \hline A \\ \hline \end{array}$. The *options*, which modify various aspects of drawn matrix through PGF's key-value system, are introduced in the following sections.

By default, the matrix is centered around its label, which is aligned with the surrounding text. The label is typeset in the surrounding mode and style.

```

 $\$$ \drawmatrix A$:  $\begin{array}{|c|} \hline A \\ \hline \end{array}$ 
{\bf \drawmatrix A}:  $\begin{array}{|c|} \hline \mathbf{A} \\ \hline \end{array}$ 
{\large \drawmatrix A}:  $\begin{array}{|c|} \hline \text{\Large A} \\ \hline \end{array}$ 

```

In equations, parentheses (spanned with `\left` and `\right`), subscripts, and superscripts naturally extend to the drawn shape: $\left(\begin{array}{|c|} \hline A \\ \hline \end{array} \subscript{i} + \begin{array}{|c|} \hline B \\ \hline \end{array} \superscript{-1} \right) \begin{array}{|c|} \hline C \\ \hline \end{array}$.

Used in matrix products such as $\begin{array}{|c|} \hline A \\ \hline \end{array} \begin{array}{|c|} \hline B \\ \hline \end{array}$, a little space (`\;`) helps to yield a more natural result: $\begin{array}{|c|} \hline A \\ \hline \end{array} \begin{array}{|c|} \hline B \\ \hline \end{array}$.

2.1 Size

By default, matrices are 1×1 large in terms of TikZ units. The width and height of a matrix are set through, respectively, `width= $\langle dimension \rangle$` and `height= $\langle dimension \rangle$` . A width or height of 0 are useful to represent vectors:

`\drawmatrix[width=0]A:` 

`size= $\langle dimension \rangle$` sets both the width and height to $\langle dimension \rangle$, resulting in a square matrix.

2.2 Shape

By default matrices are rectangular.

2.2.1 Triangular and Trapezoidal Matrices

Lower and upper triangular matrices are obtained by, respectively, setting the keys `lower` and `upper`. Hereby, non-square matrices become trapezoidal.

`\drawmatrix[lower]L:`  `\drawmatrix[upper, width=1.5]U:` 

2.2.2 Banded Matrices

Matrices are drawn as banded through the key `banded`. The band width, i.e., the horizontal/vertical extent from the diagonal, is specified through `bandwidth= $\langle dimension \rangle$` (default: 0.3);

`\drawmatrix[banded]B:`  `\drawmatrix[bandwidth=.5]B:` 

Banding for the lower and upper part of the matrices can be specified separately through `lower banded` and `upper banded`. Separate bandwidths are set through `lower bandwidth= $\langle dimension \rangle$` and `upper bandwidth= $\langle dimension \rangle$` :

`lower bandwidth` `upper bandwidth` `\drawmatrix[lower banded]B:`  `\drawmatrix[lower bandwidth=.5, upper bandwidth=.2]B:` 

Banding on rectangular matrices applies to the smaller of the two dimensions:

```
\drawmatrix[banded, width=.8]B: 
```

```
\drawmatrix[upper banded, height=.7]B: 
```

`banded` can be combined with `lower` or `upper` to draw the intersection of both shapes.

```
\drawmatrix[banded, lower]L: 
```

2.2.3 Diagonal Matrices

`diag` is a shorthand for `banded` with `bandwidth=0`:

```
\drawmatrix[diag]D: 
```

2.3 Colors and Style

By default, matrices are drawn in gray and filled white. The `TikZ` keys `draw=<color>` and `fill=<color>` change these colors. In fact, all keys not recognized by this package are passed to the `TikZ` `\filldraw` command drawing the matrix.

```
\drawmatrix[fill=yellow, draw=blue]A: 
```

```
\drawmatrix[very thick, dashed]A: 
```

2.4 The Bounding Box

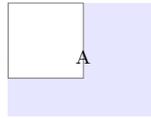
All matrices are contained in a rectangular bounding box. To draw this bounding box (e.g., to visualize the 0 entries in the matrix), use `bbox style={<style>}`; this style is applied to the `TikZ` `\node` that is the bounding box.

```
\drawmatrix[lower, bbox style={fill=blue!10}]L: 
```

By default, the bounding box is just large enough to contain the matrix. Its size is changed through the keys `bbox height=<dimension>` and `bbox width=<dimension>` (or `bbox size=<dimension>` to set them both). The label of the matrix (and thus the alignment with respect to the surrounding text) are fixed at the center of the bounding box, while the matrix is positioned at its top-left corner.

```
\drawmatrixset{bbox style={fill=blue!10}}
```

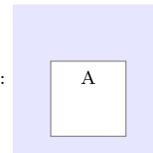
```
\drawmatrix[bbox width=2, bbox height=1.5]A:
```



The matrix can be positioned within its bounding box through `offset height= $\langle dimension \rangle$` and `offset width= $\langle dimension \rangle$` (or just `offset= $\langle dimension \rangle$` to shift along the diagonal).

```
\drawmatrixset{bbox style={fill=blue!10}}
```

```
\drawmatrix[bbox size=2, offset width=.5, offset height=.75]A:
```



2.5 Coordinate system transformations

`scale= $\langle factor \rangle$` scales all dimensions passed to a matrix:

```
\drawmatrix[scale=.6]A \drawmatrix[scale=.6, width=.5]B: A B
```

`x= $\langle value \rangle$` and `y= $\langle value \rangle$` define the coordinate system for all unit-less dimensions.

```
\drawmatrix[x=.6cm, y=.4cm]A \drawmatrix[x=.6cm, y=.4cm, width=1cm]B: A B
```

2.6 Position of the Label and Baseline

By default, the label's mid is positioned at the bounding box's center and its base is used as the whole drawing's baseline. This is controlled by the keys `label anchor= $\langle anchor \rangle$` , `label pos= $\langle position \rangle$` , and `baseline= $\langle position \rangle$` . Here, $\langle position \rangle$ has to be an anchor of one of the following nodes: `bbox` (the bounding box), `matrix` (the matrix itself), or `label` (the label).

```
\drawmatrixset{bbox height=1, height=.5, bbox style={fill=blue!10}}
```

```
\drawmatrix[label pos=matrix.north west]A: A
```

```
\drawmatrix[label pos=matrix.north west]A: A
```

```
\drawmatrix[baseline=label.north]A:
```

```
\drawmatrix[baseline=matrix.north]A:
```

```
\drawmatrix[baseline=bbox.south]A:
```

3 Changing Defaults

`\drawmatrixset` Specifying $\langle options \rangle$ with `\drawmatrixset{\langle options \rangle}` applies them to all following uses of `\drawmatrix` within the current scope.

```
\drawmatrixset{height=.5, lower}
$\drawmatrix A \; \drawmatrix B$:
```



Furthermore, TikZ keys for the entire picture, the bounding box, the matrix itself and the label can be set through the styles `every bbox`, `every drawmatrix`, and `every label`.

```
\drawmatrixset{every drawmatrix/.append style={rounded corners=5pt}}
$\drawmatrix A \; \drawmatrix[lower]B$:
```



4 Externalization

`\drawmatrix` behaves as any other TikZ picture, therefore when externalization is enabled, all matrix visualizations are also externalized. However, since there are usually many `\drawmatrix` pictures, each of which is very small and fast to produce, their externalization would mean a tremendous overhead. To avoid this overhead without explicitly dis- and re-enabling externalization throughout the document, `externalize=false` disables externalization for all `\drawmatrix` pictures:

```
\drawmatrixset{externalize=false}
```

5 Implementation

This section describes the implementation details of the `drawmatrix` package.

5.1 Package: TikZ

The `tikz` package is used for drawing.

```
1 \RequirePackage{tikz}
```

5.2 If for externalization

\TeX `if` representing whether to explicitly disable TikZ externalization.

```
\ifdrawmatrix@externalize
```

```
2 \newif\ifdrawmatrix@externalize
```

5.3 Key Declarations and Defaults

We rely on PGF keys as much as we can.

	3	<code>\pgfkeys{</code>	
		<code>Everything happens in the path /drawmatrix.</code>	
	4	<code>drawmatrix/.is family,</code>	
	5	<code>drawmatrix/.cd,</code>	
picture			<code>picture</code> is the style for the <code>\tikzpicture</code> in which the matrix is drawn. <code>baseline</code>
baseline			sets the baseline of the picture to a named coordinate of the matrix (default: base of the label).
	6	<code>picture/.style={},</code>	
	7	<code>path/.style={},</code>	
	8	<code>baseline/.style={picture/.append style={baseline=(drawmatrix #1)}},</code>	
	9	<code>scale/.style={path/.append style={scale=#1}},</code>	
	10	<code>x/.style={path/.append style={x=#1}},</code>	
	11	<code>y/.style={path/.append style={y=#1}},</code>	
	12	<code>baseline=label.base,</code>	
bbox			<code>bbox</code> is the style of the bounding box, to which <code>bbox style</code> appends keys.
bbox style	13	<code>bbox/.style={},</code>	
	14	<code>bbox style/.style={bbox/.append style={#1}},</code>	
bbox height			<code>bbox height</code> and <code>bbox width</code> don't have default values. <code>bbox size</code> sets them
bbox width			both to the same value.
bbox size	15	<code>bbox height/.initial,</code>	
	16	<code>bbox width/.initial,</code>	
	17	<code>bbox size/.style={bbox height=#1, bbox width=#1},</code>	
offset height			<code>offset height</code> and <code>offset width</code> are 0 by default. <code>offset</code> sets them both to
offset width			the same value.
offset	18	<code>offset height/.initial=0,</code>	
	19	<code>offset width/.initial=0,</code>	
	20	<code>offset/.style={offset height=#1, offset width=#1},</code>	
height			<code>width</code> and <code>height</code> are 1 (TikZ unit) by default. <code>size</code> sets them both to the same
width			value.
size	21	<code>height/.initial=1,</code>	
	22	<code>width/.initial=1,</code>	
	23	<code>size/.style={height=#1, width=#1},</code>	
lower bandwidth			The <code>lower bandwidth</code> and <code>upper bandwidth</code> don't have default values.
upper bandwidth			<code>bandwidth</code> sets them both to the same value.
bandwidth	24	<code>lower bandwidth/.initial,</code>	
	25	<code>upper bandwidth/.initial,</code>	
	26	<code>bandwidth/.style={lower bandwidth=#1, upper bandwidth=#1},</code>	

lower banded **upper banded** **banded** **lower** **upper** **diag** **externalize** **label** **label pos** **label anchor**

lower banded and **upper banded** are shortcuts to set the corresponding bandwidths to the default value of 0.3 (TikZ units). **banded** sets them both.

```

27 lower banded/.style={lower bandwidth=.3},
28 upper banded/.style={upper bandwidth=.3},
29 banded/.style={lower banded, upper banded},

```

lower and **upper** are implemented by setting the opposite bandwidth to 0. **diag** sets them both.

```

30 lower/.style={upper bandwidth=0},
31 upper/.style={lower bandwidth=0},
32 diag/.style={lower, upper},

```

externalize sets a \TeX if (default: true = behave as all pictures).

```

33 externalize/.is if=drawmatrix@externalize,
34 externalize=true,

```

label is the style for the label. **label pos** sets the label at a named coordinate of the matrix (default: center of the bounding box). **label anchor** sets the label's anchor (default: in the middle).

```

35 label/.style={},
36 label pos/.style={label/.append style={at=(drawmatrix #1)}},
37 label pos=bbbox.center,
38 label anchor/.style={label/.append style={anchor=#1}},
39 label anchor=mid,

```

Unknown keys are collected in `/drawmatrix/drawmatrix`.

```

40 drawmatrix/.style={},
41 .unknown/.code={%
42   \let\dm@currname\pgfkeyscurrentname%
43   \let\dm@currval\pgfkeyscurrentvalue%
44   \ifx#1\pgfkeysnovalue\pgfkeysalso{
45     drawmatrix/.append style/.expand once={\dm@currname}
46   }\else\pgfkeysalso{
47     drawmatrix/.append style/.expand twice={%
48       \expandafter\dm@currname\expandafter=\dm@currval%
49     }
50   }\fi%
51 },

```

The default style for matrices: **every picture** applies to all `\tikzpictures` the matrices are drawn in, **every bbox** applies to all bounding boxes, **every drawmatrix** applies to the matrices themselves, and **every label** applies to the labels.

```

52 every picture/.style={},
53 every bbox/.style={
54   name=drawmatrix bbox,
55   inner sep=0,
56 },

```

```

57 every drawmatrix/.style={
58     fill=white,
59     draw=gray,
60 },
61 every label/.style={
62     name=drawmatrix label,
63     outer sep=0,
64     inner sep=0,
65 },
66 every node/.style={
67     name=drawmatrix matrix,
68     outer sep=0,
69     inner sep=0,
70     anchor=north west,
71 }
72 }

```

5.4 User Macros

`\drawmatrixset` as a simple shortcut like `\tikzset`.

```
\drawmatrixset
```

```
73 \newcommand\drawmatrixset[1]{\pgfqkeys{/drawmatrix}{#1}}
```

Here we go, the main thing: `\drawmatrix`. First, apply the options and extract the sizes from the PGF keys.

```
\drawmatrix
```

```

74 \newcommand\drawmatrix[2] [] { {%
75     \drawmatrixset{
76         #1,
77         height/.get=\dm@height,
78         width/.get=\dm@width,
79         lower bandwidth/.get=\dm@lowerbandwidth,
80         upper bandwidth/.get=\dm@upperbandwidth,
81         offset height/.get=\dm@offsetheight,
82         offset width/.get=\dm@offsetwidth,
83         bbox height/.get=\dm@bboxheight,
84         bbox width/.get=\dm@bboxwidth,
85     }%

```

Prepare the label text (keep math mode).

```
86     \ifmmode\def\dm@labeltext{ $#2$ }\else\def\dm@labeltext{ #2 }\fi%
```

Disable externalization if `externalize=false`. Start the picture.

```

87     \ifdrawmatrix@externalize\else%
88         \ifx\tikz@library@external@loaded\undefined\else%
89             \tikzset{external/export=false}%
90         \fi%
91     \fi%
92     \begin{tikzpicture} [/drawmatrix/every picture, /drawmatrix/picture]

```

Parse width, height, the minimum dimension and zero for comparison purposes.

```

93     \path[/drawmatrix/path] (\dm@width, \dm@height);
94     \pgfgetlastxy\dm@width\dm@height
95     \path[/drawmatrix/path] (\dm@offsetwidth, \dm@offsetheight);
96     \pgfgetlastxy\dm@offsetwidth\dm@offsetheight
97     \pgfmathsetlengthmacro\dm@minsize{min(\dm@width, \dm@height)}
98     \pgfmathsetlengthmacro\dm@zero{0.0}

```

Prepare the band widths: First, if the matrix is not banded, the bandwidth is set to the smaller matrix dimension. Then, the band width is limited by this smaller dimension.

```

99     \expandafter\ifx\dm@lowerbandwidth\pgfkeysnovalue
100     \edef\dm@lowerbandwidth{\dm@minsize}
101     \else
102     \path[/drawmatrix/path] (\dm@lowerbandwidth, 0);
103     \pgfgetlastxy\dm@lowerbandwidth\dm@zero
104     \fi
105     \expandafter\ifx\dm@upperbandwidth\pgfkeysnovalue
106     \edef\dm@upperbandwidth{\dm@minsize}
107     \else
108     \path[/drawmatrix/path] (0, \dm@upperbandwidth);
109     \pgfgetlastxy\dm@zero\dm@upperbandwidth
110     \fi
111     \pgfmathsetlengthmacro\dm@lowerbandwidth{
112     min(\dm@minsize, \dm@lowerbandwidth)
113     }
114     \pgfmathsetlengthmacro\dm@upperbandwidth{
115     min(\dm@minsize, \dm@upperbandwidth)
116     }

```

Set the default bounding box size.

```

117     \expandafter\ifx\dm@bboxheight\pgfkeysnovalue
118     \pgfmathsetlengthmacro\dm@bboxheight{
119     \dm@height + \dm@offsetheight
120     }
121     \else
122     \path[/drawmatrix/path] (0, \dm@bboxheight);
123     \pgfgetlastxy\dm@zero\dm@bboxheight
124     \fi
125     \expandafter\ifx\dm@bboxwidth\pgfkeysnovalue
126     \pgfmathsetlengthmacro\dm@bboxwidth{
127     \dm@width + \dm@offsetwidth
128     }
129     \else
130     \path[/drawmatrix/path] (\dm@bboxwidth, 0);
131     \pgfgetlastxy\dm@bboxwidth\dm@zero
132     \fi

```

Reset the bounding box and begin with (drawing) the path for the bounding box.

```

133     \pgfresetboundingbox
134     \node[/drawmatrix/every bbox, /drawmatrix/bbox,
135           minimum height=\dm@bboxheight,
136           minimum width=\dm@bboxwidth] {};

```

Whether needed or not, declare all matrix corners.

```

137     \path (drawmatrix bbox.north west)
138           ++(\dm@offsetwidth, -\dm@offsetheight)
139           ++(.5\pgflinewidth, -.5\pgflinewidth)
140           coordinate (drawmatrix north west)
141           ++(\dm@width, 0)
142           +(-\dm@minsize + \dm@upperbandwidth, 0)
143           coordinate (drawmatrix north)
144           +(0, -\dm@minsize + \dm@upperbandwidth)
145           coordinate (drawmatrix east)
146           ++(0, -\dm@height)
147           coordinate (drawmatrix south east)
148           ++(-\dm@width, 0)
149           +(\dm@minsize - \dm@lowerbandwidth, 0)
150           coordinate (drawmatrix south)
151           +(0, \dm@minsize - \dm@lowerbandwidth)
152           coordinate (drawmatrix west);

```

Add an invisible node the size of the matrix.

```

153     \node[/drawmatrix/every node,
154           minimum height=\dm@height,
155           minimum width=\dm@width]
156     at (drawmatrix north west) {};

```

Now, draw only what is needed of the matrix. Otherwise path modifications (e.g., such as rounded corners) might not work.

```

157     \filldraw[/drawmatrix/every drawmatrix, /drawmatrix/drawmatrix]
158     (drawmatrix north west)
159     \ifx\dm@upperbandwidth\dm@zero
160       \ifx\dm@width\dm@minsize\else -- (drawmatrix north) \fi
161       \ifx\dm@height\dm@minsize\else -- (drawmatrix east) \fi
162     \else
163       -- (drawmatrix north)
164       \ifx\dm@upperbandwidth\dm@minsize\else
165         -- (drawmatrix east)
166       \fi
167     \fi
168     -- (drawmatrix south east)
169     \ifx\dm@lowerbandwidth\dm@zero
170       \ifx\dm@width\dm@minsize\else -- (drawmatrix south) \fi
171       \ifx\dm@height\dm@minsize\else -- (drawmatrix west) \fi
172     \else
173       -- (drawmatrix south)
174       \ifx\dm@lowerbandwidth\dm@minsize\else
175         -- (drawmatrix west)
176       \fi

```

```

177          \fi
178          -- cycle;

    The label.
179          \node[/drawmatrix/every label, /drawmatrix/label]
180              {\dm@labeltext};
181      \end{tikzpicture}%
182 }}

```

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

B	<code>\every_picture</code> 52	<code>\offset_height</code> 4 , 18
<code>\banded</code> 3 , 27	<code>\externalize</code> 5 , 33	<code>\offset_width</code> 4 , 18
<code>\bandwidth</code> 3 , 24		
<code>\baseline</code> 5 , 6	H	P
<code>\bbox</code> 13	<code>\height</code> 2 , 21	<code>\picture</code> 6
<code>\bbox_height</code> 4 , 15	I	
<code>\bbox_size</code> 4 , 15	<code>\ifdrawmatrix@externalize</code>	S
<code>\bbox_style</code> 4 , 13 2	<code>\size</code> 2 , 21
<code>\bbox_width</code> 4 , 15	L	
D	<code>\label</code> 35	U
<code>\diag</code> 3 , 30	<code>\label_anchor</code> 5 , 35	<code>\undefined</code> 88
<code>\drawmatrix</code> 1 , 74	<code>\label_pos</code> 5 , 35	<code>\upper</code> 2 , 30
<code>\drawmatrixset</code> 5 , 73 , 75	<code>\lower</code> 2 , 30	<code>\upper_banded</code> 3 , 27
E	<code>\lower_banded</code> 3 , 27	<code>\upper_bandwidth</code> 3 , 24
<code>\every_bbox</code> 5 , 52	<code>\lower_bandwidth</code> 3 , 24	
<code>\every_drawmatrix</code> 5 , 52	O	W
<code>\every_label</code> 5 , 52	<code>\offset</code> 4 , 18	<code>\width</code> 2 , 21

Change History

v1.0.0	linewidth/2 offset 1
General: Initial Version 1	v1.1.1
v1.0.1	General: Bugfix: Remove extra
General: Bugfix: Collapsible bbox	space after vectors 1
(label placement for vectors) 1	v1.2.0
v1.0.2	General: Add coordinate
General: Bugfix: Bbox had a	transformations 1